

**IN THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application.

**Listing of Claims**

1. (Previously Presented): A method of processing frames for a TCP connection, comprising:
  - processing a first portion of the frames using an offload unit to produce first processed frame data;
  - uploading the first processed frame data to a user buffer in a first portion of memory that is allocated to an application program;
  - processing a second portion of the frames using the offload unit to produce second processed frame data;
  - uploading the second processed frame data to a legacy buffer in a second portion of the memory that is allocated to a software driver configured to communicate between the offload unit and a TCP stack; and
  - processing the second processed frame data using the TCP stack executed on a CPU to produce third processed frame data.
2. (Original): The method of claim 1, further comprising determining whether a special case exists during the processing of each of the frames.
3. (Original): The method of claim 1, wherein at least a portion of the first processed frame data is payload data.
4. (Original): The method of claim 1, wherein at least a portion of the second processed frame data is partially processed frame header data.
5. (Original): The method of claim 1, wherein at least a portion of the second processed frame data is payload data.

6. (Cancelled)
7. (Cancelled)
8. (Original): The method of claim 1, further comprising finishing processing of the second processed frame data by the TCP stack executed on the CPU.
9. (Previously Presented): A system for processing data for a TCP connection, comprising:
  - a TCP stack configured to process received frames stored in at least one legacy buffer;
  - a software driver configured to interface between the TCP stack and an offload unit;
  - a memory configured to store user buffers in a first portion that is allocated to an application program and to store legacy buffers in a second portion that is allocated to the software driver; and
  - the offload unit configured to:
    - process frames received on a delegated connection to produce payload data and partially processed frames;
    - upload partially processed frames to at least one of the legacy buffers; and
    - upload the payload data to at least one of the user buffers.
10. (Original): The system of claim 9, wherein the offload unit is configured to process frames for which a special case does not exist.
11. (Original): The system of claim 9, wherein the offload unit is configured to notify the TCP stack when a special case is determined to exist.
12. (Original): The system of claim 9, wherein the TCP stack is configured to process frames for which a special case exists.

13. (Cancelled)
14. (Cancelled)
15. (Previously Presented): The system of claim 9, wherein the offload unit is configured to receive additional frames while uploading payload data to one of the legacy buffers or uploading processed frames to one of the user buffers.
16. (Previously Presented): A method of processing frames for delegated and non-delegated TCP connections, comprising:
  - processing delegated TCP connections using an offload unit, the offload unit configured to process frames for which special cases do not exist to produce processed frame data;
  - uploading the processed frame data to a user buffer in a first portion of memory that is allocated to an application program;
  - processing non-delegated TCP connections using a TCP stack executing on a CPU;
  - uploading frame data for the non-delegated TCP connections to a legacy buffer in a second portion of the memory that is allocated to a software driver configured to communicate between the offload unit and the TCP stack; and
  - processing all frames for which special cases exist using the TCP stack executing on the CPU.
17. (Original): The method of claim 16, wherein the processing of the delegated TCP connections produces payload data.
18. (Cancelled)

19. (Original): The method of claim 16, further comprising updating connection state information stored in the offload unit when data is received for a delegated connection.
20. (Original): The method of claim 16, further comprising updating connection state information stored in the offload unit when data is transmitted for a delegated connection.
21. (Previously Presented): The method of claim 1, further comprising issuing an interrupt to the CPU after the second processed frame data is uploaded to the legacy buffer.
22. (Previously Presented): The system of claim 9, wherein the user buffers are stored in physically contiguous memory locations within the first portion of the memory.
23. (Previously Presented): The system of claim 9, wherein the legacy buffers are stored in physically contiguous memory locations within the second portion of the memory.
24. (Previously Presented): The system of claim 9, wherein the user buffers are stored in physically non-contiguous memory locations within the first portion of the memory.
25. (Previously Presented): The system of claim 9, wherein the legacy buffers are stored in physically non-contiguous memory locations within the second portion of the memory.